Input and Output

Character input and output :

A] getchar() and putchar() :-

1) getchar()

   This function reads the next available character from the screen and returns it as an integer. This function reads only single character at a time. You can use this method in the loop in case you want to read more than one character from the screen.
   Syntax : int getchar(void);

2) putchar()

   This function puts a character on the screen and returns the same character. This function puts only single character at a time. You can use this method in the loop in case you want to display more than one character on the screen.
   Syntax : int putchar(int c);

Example :
#include <stdio.h>
main()
{
    char c;
    printf("Enter a value ");
    c = getchar();
    printf("\nYou entered: ");
    putchar( c );
}

When the above code is compiled and executed, it waits for you to input some text. When you enter a text and press enter, then the program proceeds and reads only a single character and displays it as follows:

$./a.out
Enter a value : this is test
You entered: t

B] getch() and putch() :-

1) getch()

   This function is used to get character from console that is from keyboard without echo to screen.
   Syntax : int getch(void);

2) `putch()`

This function is used to display all alphanumeric characters through the standard output device like monitor. This function displays single character at a time.

Syntax: `int putch(int);`

Example:
```
#include<stdio.h>
#include<conio.h>

{ 
  char c;
  clrscr();
  c=getch();
  putch(c);
}
```

C] `getche()`

This function is used to get character from console with echo to the screen.

Syntax: `int getche(void);`

String input and output:

1) `gets()` :-
This function reads a line from `stdin` into the buffer pointed to by `s` until either a terminating newline or EOF (End of File).

Syntax: `char *gets(char *s)`

2) `puts()` :-
This function writes the string 's' to `stdout`. It appends new line character at the end of the string.

Syntax: `int puts(const char *s)`

Example:
```
#include <stdio.h>
int main()
{
  char str[100];
```
printf("Enter a value : ");
gets(str);
printf("You entered: ");
puts(str);
return 0;
}

When the above code is compiled and executed, it waits for you to input some text. When you enter a text and press enter, then the program proceeds and reads the complete line till end, and displays it as follows:

$./a.out
Enter a value : this is test
You entered: this is test

**Formatted input and output functions:**

The stdio.h header file provides built-in functions for reading and writing formatted data from input output device.

1) **Formatted Input [ scanf() ] :-**

This function reads the input from the standard input stream stdin and scans that input according to the format provided.

Syntax: int scanf(const char *format, &var1,&var2.....&varn);

The format can be a simple constant string, but you can specify %s, %d, %c, %f, etc., to print or read strings, integer, character, or float, respectively. There are many other formatting options available which can be used based on requirements.

Format consist of
1. White space characters
2. Conversion characters
3. Printable characters which specifies delimiter in between two characters
4. Modifiers which tells size and type of data item.

2) **Formatted Output [ printf() ] :-**

This function writes the output to the standard output stream stdout and produces the output according to the format provided.

Syntax: int printf(const char *format, var1,var2.....varn);

Format consist of
1. Ordinary characters
2. Conversion characters
3. Escape Sequences
```c
#include <stdio.h>
int main()
{
    char str[100];
    int i;
    printf("Enter a value :\n");
    scanf("%s %d", str, &i);
    printf("You entered: %s %d \n", str, i);
    return 0;
}
```

When the above code is compiled and executed, it waits for you to input some text. When you enter a text and press enter, then program proceeds and reads the input and displays it as follows:

```
$.a.out
Enter a value : hello 123
You entered: hello 123
```

Here, it should be noted that scanf() expects input in the same format as you provided %s and %d, which means you have to provide valid inputs like "string integer". If you provide "string string" or "integer integer", then it will be assumed as wrong input. Secondly, while reading a string, scanf() stops reading as soon as it encounters a space, so "this is test" are three strings for scanf().

**Solved Programs :**

**Q.1) Write a program to calculate area and circumference of a circle.**

```c
#include<stdio.h>
#define PI 3.14
main()
{
    float r,area,cir;
    printf("Enter radius :\n");
    scanf("%f", &r);
    area=PI*r*r;
    cir=2*PI*r;
    printf("Area = %f
Circumference = %f",area,cir);
}
```

**Q.2) Write a program to accept a character and print its ASCII value.**

```c
#include<stdio.h>
main()
{
    char ch;
    printf("Enter a character :\n");
    scanf("%c", &ch);
    printf("ASCII value of %c is %d",ch,ch);
}
```
Q.3) Write a program to accept two numbers and interchange them by using third variable.
#include<stdio.h>
main()
{
    int x,y,temp;
    printf("Enter two numbers : ");
    scanf("%d%d", &x,&y);
    temp=x;
    x=y;
    y=temp;
    printf("Interchanged numbers are %d and %d", x,y);
}

Q.4) Write a program to accept two numbers and interchange them without using third variable.
#include<stdio.h>
main()
{
    int x,y;
    printf("Enter two numbers : ");
    scanf("%d%d", &x,&y);
    x=x+y;
    x=x-y;
    y=x-y;
    printf("Interchanged numbers are %d and %d", x,y);
}